Curriculum Learning of Bayesian Network Structures

Yanpeng Zhao[1], Yetian Chen[2], Kewei Tu[1] and Jin Tian[2]

[1]School of Information Science and Technology, ShanghaiTech University, Shanghai, China
[2]Department of Computer Science, Iowa State University, Ames, IA, USA

Introduction

- Bayesian Network (BN)
  - A directed acyclic graph (DAG) where nodes are random variables and directed edges represent probability dependencies among variables
- BN Structure Learning
  - Firstly construct the topology (structure) of the network
  - Then estimate the parameters (CPDs) given the fixed structure
- Curriculum Learning (CL) [*Yoshua Bengio et al. ICML 2009*]
  - **Ideas:** learn with the simpler samples or easier tasks as the start
  - **Definition:** a curriculum is a sequence of weighting schemes of the training data $\langle W_1, W_2, \dots, W_n \rangle$, where W_1 assigns more weight to easier samples, then each next scheme assigns more weight to harder samples, at last W_n assigns uniform weight to all samples

Learn BN Structure via CL

- Motivation
  - Given a set of variables, human rarely try to find the dependency relations between all variables by looking at all the training samples at once
  - Instead, human learn in a more organized way, starting with more common samples that involve dependency relations between only a small subset of variables

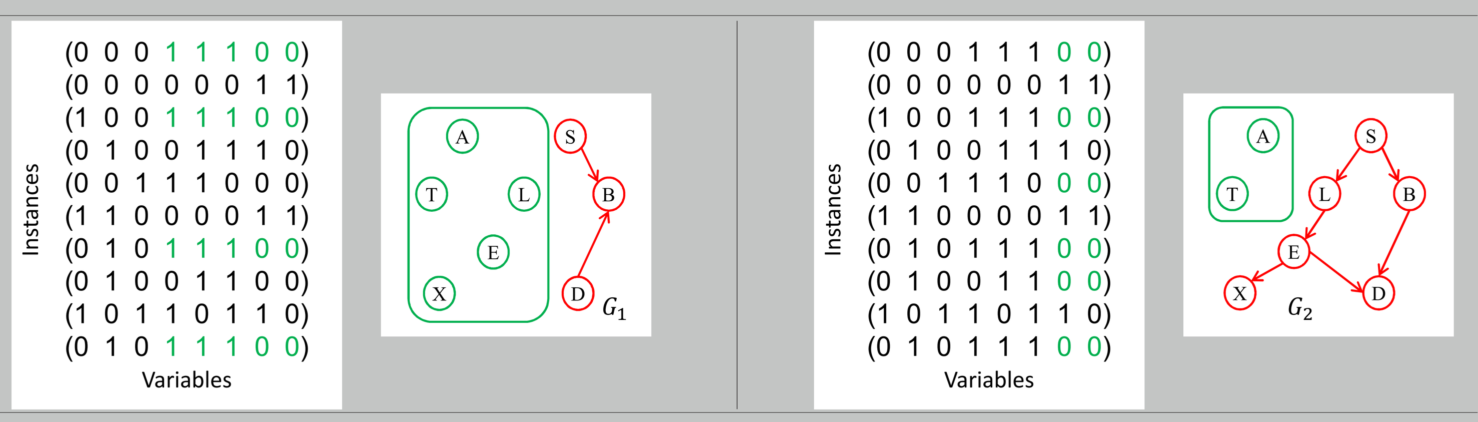
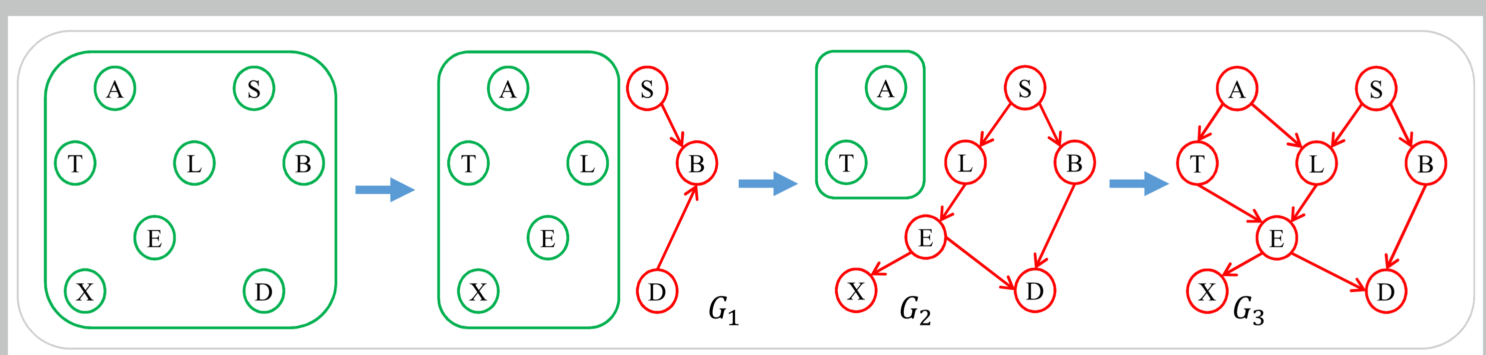


Figure 1 : Variables S, B, D, L, E, X, A, T correspond to each column of the dataset respectively. **Left:** at stage 1, learn a subnet G_1 over $\{S, B, D\}$ from scratch with the rest variables fixed at (1 1 1 0 0); **Right:** at stage 2, learn a larger subnet G_2 over $\{S, B, D, L, E, X\}$ with G_1 as the start point of search while fix the rest variables at (0 0)

- Curriculum in BN Structure Learning

We define the curriculum as $(\mathrm{X}_{(1)}, \dots, \mathrm{X}_{(n)})$, *a sequence of selected subsets of the random variables* $\mathrm{X}_{(i)}$, *over which the corresponding subnet* G_i *is learnt.*

Where $\mathrm{X} = (X_1, \dots, X_n)$ is a *variable set*, $\mathrm{X}_{(i)} \subseteq \mathrm{X}$, $\mathrm{X}'_{(i)} = \mathrm{X} \setminus \mathrm{X}_{(i)}$, $\mathrm{X}_{(i)} \subset \mathrm{X}_{(i+1)}$. $(G_1, \dots, G_m)$ is a sequence of *intermediate learning targets.*



Limitation and Solution

- Limitation
  - We only used a small fraction of the dataset at each learning stage
- Solution

Let $\mathrm{X}'_{(i)}$ *take value from* $\{\mathrm{x}'_{(i),1}, \dots, \mathrm{x}'_{(i),q}\}$, *the set of data segments* $D_i = \{D_{i,1}, \dots, D_{i,q}\}$ *by grouping samples based on the values of* $\mathrm{X}'_{(i)}$.

  - **An Important Observation:** when we fix $\mathrm{X}'_{(i)}$ to different values, our learning target is actually the same DAG structure G_i but with different parameters (CPDs)
  - **Assumption:** $D_{i,1}, \dots, D_{i,q}$ are generated by the same G_i but with *independent* CPDs
  - We can revise the scoring function to take into account multiple versions of parameters

Algorithm

Algorithm 1: Curriculum Learning of BN Structures

input: Variable Set X, Training Data D, Curriculum $(\mathrm{X}_{(1)}, \dots, \mathrm{X}_{(m)})$. G_0 is initialized to a network containing variables in $\mathrm{X}_{(1)}$ with no edge.
for $i \dots m$ **do**
  Generate the set of data segments $D_i = \{D_{i,1}, \dots, D_{i,q}\}$ based on the values of $\mathrm{X} \setminus \mathrm{X}_{(i)}$
  $G_i \leftarrow search(D_i, \mathrm{X}_{(i)}, G_{i-1})$
end
return: G_m

Scoring Function

- Bayesian Score Function

$$\log P(G_i, D_i) = C + \sum_{j=1}^{q} \log P(G_i, D_{i,j})$$

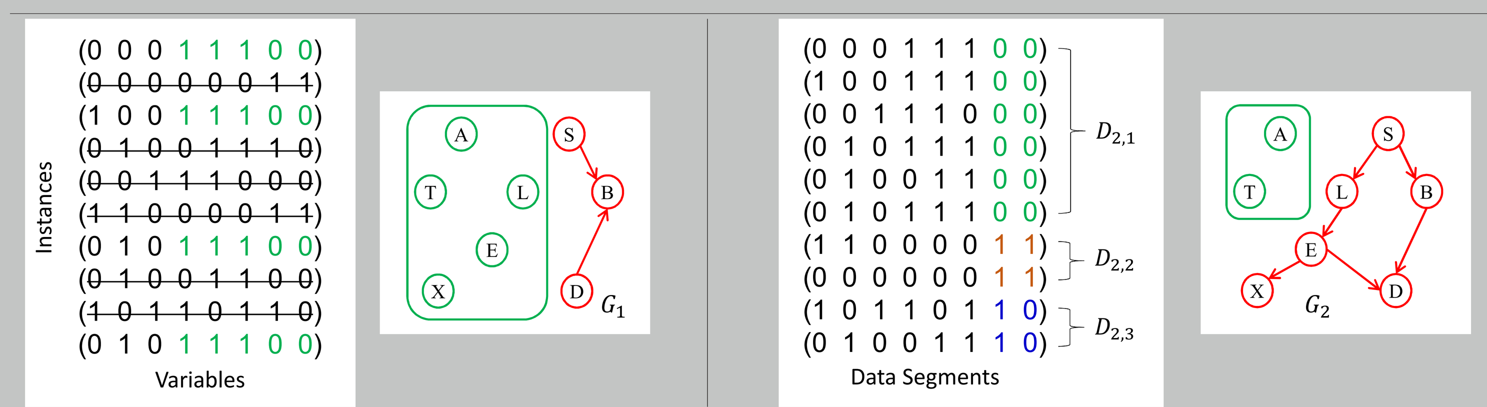


Figure 2 : **Left:** using the previous method, learn a subnet G_1 over $\mathrm{X}_{(1)} = \{S, B, D\}$. We only used samples with $\mathrm{X}'_{(1)} = \{L, E, X, A, T\}$ fixed at (1 1 1 0 0), the samples with a strikeout **IS NOT** used; **Right:** using the new method, learn a subnet G_2 over $\mathrm{X}_{(2)} = \{S, B, D, L, E, X\}$. $\mathrm{X}'_{(2)}$ takes value from $\{(0,0), (1,1), (1,0)\}$, we divide the dataset into three partitions by grouping samples based on the values of X' and use all of them

- Penalty Term

Over-fitting occurs when sample size is small or there are many stages, so we use a penalty term

$$penalty(G_i : D_i) = \left(\frac{a}{SS} + \frac{V(G_i)}{b}\right) E(G_i),$$

SS: sample size; $V(G_i)$: number of the variables in G_i, $E(G_i)$: number of edges in G_i; a, b: positive constants.

- The Final Score Function

$$score(G_i : D_i) = \log P(G_i, D_{i,j}) - penalty(G_i : D_i)$$

Theorems

Theorem 1 . *For any* i, j, k *s.t.* $1 \le i < j < k \le n$, *we have*

$$d_H(G_i, G_k) \ge d_H(G_j, G_k)$$

where $d_H(G_i, G_j)$ *is the structural Hamming distance (SHD) between the structures of two BNs* G_i *and* G_j.

Theorem 2 . *For any* i, j, k *s.t.* $1 \le i < j < k \le n$, *we have*

$$d_{TV}(G_i, G_k) \ge d_{TV}(G_j, G_k)$$

where $d_{TV}(G_i, G_j)$ *is the total variation distance between the two distributions defined by the two BNs* G_i *and* G_j.

Experiments

- 10 benchmark BNs from the *bnlearn* repository (alarm, andes, asia, child, hailfinder, hepar2, insurance, sachs, water, win95pts)
- Comparisons with MMHC [*Ioannis Tsamardinos et al. ML 2006*] under metrics of BDeu, BIC, KL and SHD

		Sample Size (SS)					
Metric	Algorithm	100	500	1000	5000	10000	50000
BDeu	CL	1(0)	**1(10)**	**1(9)**	**1(8)**	**1(10)**	**1(8)**
	MMHC	**0.89(10)**	1.06(0)	1.02(1)	1.01(2)	1.02(0)	1.01(2)
BIC	CL	1(0)	**1(9)**	**1(9)**	**1(6)**	**1(8)**	**1(8)**
	MMHC	**0.88(10)**	1.07(1)	1.02(1)	1.02(4)	1.02(2)	1.01(2)
KL	CL	1(0)	**1(10)**	**1(9)**	**1(7)**	**1(9)**	**1(9)**
	MMHC	**1.71(10)**	0.82(0)	0.96(1)	0.96(2)	0.97(0)	0.97(0)
SHD	CL	**1(7)**	**1(9)**	**1(7)**	**1(7)**	**1(8)**	**1(6)**
	MMHC	1.06(3)	1.26(1)	1.29(3)	1.07(2)	1.21(1)	1.24(3)

- Verification of Theorem 1

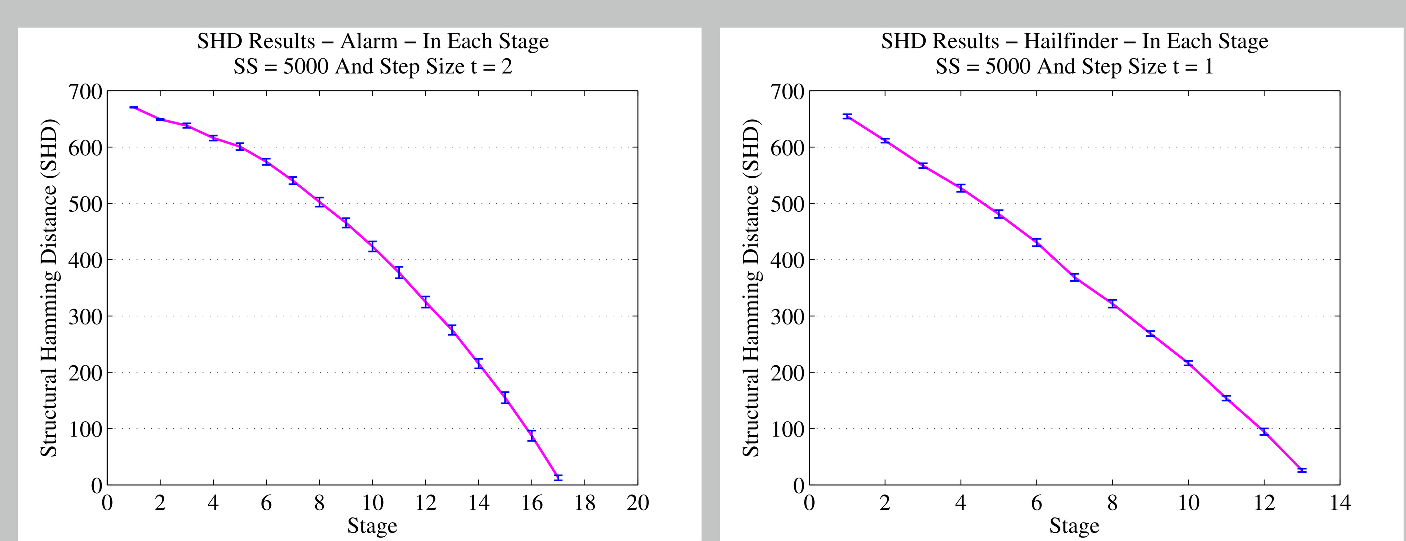


Figure 3 : SHD between the intermediate learning result at each stage and the target BN

Conclusions

- We proposed a curriculum learning algorithm for BN structure learning
- We tailored the bayesian scoring function for our algorithm
- We proved two theorems that show theoretical properties of our algorithm
- We empirically showed that our algorithm outperformed the state-of-the-art MMHC algorithm in learning BN structures